



“DESARROLLO DE PROYECTO SEMILLA PARA TIENDAS VIRTUALES
BASADO EN TECNOLOGÍA DJANGO”

Diego Alejandro Martínez Chico
160003618

Universidad de los Llanos
Facultad de Ciencias Básicas e Ingeniería
Escuela de ingeniería
Programa de Ingeniería de Sistemas
Villavicencio, Colombia
Febrero 2022



**“DESARROLLO DE PROYECTO SEMILLA PARA TIENDAS VIRTUALES
BASADO EN TECNOLOGÍA DJANGO”**

Diego Alejandro Martínez Chico
160003618

Trabajo de grado para obtener el título de Ingeniero de Sistemas

Bajo la Dirección de
Javier Mauricio Enciso Moreno
Ingeniero de Sistemas, Magíster en Ingeniería de Sistemas

Universidad de los Llanos
Facultad de Ciencias Básicas e Ingeniería
Escuela de ingeniería
Programa de Ingeniería de Sistemas
Villavicencio, Colombia
Febrero 2020

AUTORIZACIÓN

Yo Diego Alejandro Martínez Chico mayor de edad, vecino de la Ciudad de Villavicencio - Meta, identificado con la Cédula de Ciudadanía No. 1'121.912.983 de Villavicencio - Meta, actuando en nombre propio en mi calidad de autor del trabajo de tesis, monografía o trabajo de grado denominado “DESARROLLO DE PROYECTO SEMILLA PARA TIENDAS VIRTUALES BASADO EN TECNOLOGÍA DJANGO”, hago entrega del ejemplar y de sus anexos de ser el caso, en formato digital o electrónico (CD-ROM) y autorizo a la UNIVERSIDAD DE LOS LLANOS, para que en los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, con la finalidad de que se utilice y use en todas sus formas, realice la reproducción, comunicación pública, edición y distribución, en formato impreso y digital, o formato conocido o por conocer de manera total y parcial de mi trabajo de grado o tesis.

EL AUTOR – Diego Alejandro Martínez Chico, como autor, manifiesto que el trabajo de grado o tesis objeto de la presente autorización, es original y se realizó sin violar o usurpar derechos de autor de terceros; por tanto, la obra es de mi exclusiva autoría y poseo la titularidad sobre la misma; en caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, como autor, asumiré toda la responsabilidad, y saldré en defensa de los derechos aquí autorizados, para todos los efectos la Universidad actúa como un tercero de buena fe.

Para constancia, se firma el presente documento en dos (2) ejemplares del mismo valor y tenor en Villavicencio - Meta, a los 27 días del mes de marzo de dos mil veintidós (2022).

Diego Alejandro Martínez Chico

Firma: _____

Nombre: _____

C.C. No. _____ de _____

NOTAS DE ACEPTACIÓN

Javier Mauricio Enciso Moreno
Director

Jurado

Fecha

Índice general

AUTORIZACIÓN	3
NOTAS DE ACEPTACIÓN	4
Índice general.....	5
Índice de ilustraciones.....	3
Índice de tablas	3
1. Agradecimientos	4
2. Resumen	5
3. Planteamiento del problema	6
4. Objetivos.....	7
4.1 Objetivo General	7
4.2 Objetivo Específicos	7
5. Justificación	8
6. Marco teórico.....	9
6.1. Django.....	10
6.2. Git	10
6.2.1 Flujo de trabajo en Git.	10
6.3. Docker.....	12
6.4. Bootstrap	12
7. Metodología y área de estudio	12
7.1. Eventos de Scrum	13
8. Ejecución	14
8.1. Desarrollo de las prácticas	14
8.1.1 Descripción de la duración de la pasantía	15
8.2 Implementación	19
9. Resultados.....	20
9.1. Requerimientos	20
9.2. E-Commerce	24
9.3. Pasarela de Pagos	28
9.4. CMS	31
9.5. Despliegue.....	33
10. Discusión	34

11.	Conclusiones.....	35
12.	Recomendaciones	36
13.	Bibliografía.....	37
14.	Anexos	38

Índice de ilustraciones

Ilustración 1. Aumento del comercio electrónico del 2019 al 2021. Crédito: Kantar.....	9
Ilustración 2. Flujo de trabajo en Git. Representación visual de Gitflow. Crédito: Atlassian	11
Ilustración 3. Modelo BPMN del e-commerce. Crédito: Propio.....	15
Ilustración 4. Modelo de datos del e-commerce. Crédito: Propio.....	16
Ilustración 5. Template que se utilizará como base para las vistas del e-commerce.	17
Crédito: Propio.....	17
Ilustración 6. Listado de artículos de Enciso Systems Crédito: Propio.....	18
Ilustración 7. Arquitectura del proyecto Créditos: Propio.	24
Ilustración 8. Campos, atributos y funciones del modelo OrderItem. Crédito: Propio.	25
Ilustración 9. Resultado final del template listar productos. Crédito: Propio	25
Ilustración 10. Resultado Final del template detalle del producto. Crédito: Propio	26
Ilustración 11. Resultado final del template carrito de compras. Crédito: Propio	26
Ilustración 12. Vista final del template Login. Crédito: Propio.	27
Ilustración 13. Vista final donde un usuario puede registrarse en el e-commerce. Crédito: Propio.....	27
Ilustración 14. Vista del perfil de usuario. Crédito: Propio.....	28
Ilustración 15. Flujo Técnico que realiza el botón de pagos de PayPal. Crédito: PayPal	29
Ilustración 16. Botones de pago con PayPal. Crédito: Propio.....	29
Ilustración 17. Vista de la confirmación del pago. Crédito: Propio.	30
Ilustración 18. Visualización de los datos del pago en el administrador de Django. Crédito: Propio.....	30
Ilustración 19. Creación de una página desde el sitio de administración. Crédito: Propio	31
Ilustración 20. Lista del menú de navegación. Crédito: Propio.	32
Ilustración 21. Vista de listado de artículos de Enciso Systems. Crédito: Propio.....	32
Ilustración 22. Vista detallada de un artículo. Crédito: Propio.	33

Índice de tablas

Tabla 1. Requisitos funcionales del sitio web (E-Commerce). Crédito: Propio.....	21
Tabla 2. Requerimientos funcionales del CMS. Crédito: Propio.....	22

1. Agradecimientos

Agradezco primeramente a mis padres Cecilia Chico y Alberto Martínez, por sus consejos y enseñanzas, por inculcarme buenos valores y siempre guiarme por el camino de Dios, por el apoyo que me han dado, por el esfuerzo y la fe que depositaron en mí para poder realizar mis estudios. A mi Hermana Laura Camila Martínez que siempre estuvo apoyándome y brindándome su compañía.

Al director de mi pasantía, el ingeniero Javier Enciso, por brindarme su confianza y darme la oportunidad de realizar este proyecto.

A mis amigos y compañeros de carrera que durante estos años de estudio me ayudaron en mi crecimiento personal y profesional, por ser parte de mi proceso de aprendizaje y por apoyarme y brindarme su compañía en todo momento.

A los maestros e ingenieros por brindarme sus conocimientos, enseñanzas y experiencias durante todos estos años.

Y, ante todo, agradecerle a Dios por las bendiciones que me ha dado, por guiarme y permitirme llegar hasta este punto de mi vida.

2. Resumen

Los pasos en la evolución están guiando al comercio por nuevas direcciones, abriendo caminos en la era del internet, cambiando la forma común de vender y comprar, es así cómo aparece el comercio electrónico (E-Commerce). Por ello y con el objetivo de entrar en dicho mercado, los negocios buscan empresas de software que los ayude con el desarrollo de un sitio web donde puedan ofrecer sus productos.

Para atender esta demanda, Enciso Systems cuenta con un proyecto semilla para tiendas virtuales que permite realizar diversos proyectos. Para el desarrollo del proyecto se utilizó el framework web (Django) en el cual se desarrolló tanto el sistema de gestión de contenidos (CMS) y el E-Commerce.

El CMS cuenta con 3 aplicaciones, en las que el usuario de una forma muy fácil desde el administrador de Django puede administrar el contenido de los menús de navegación, el contenido de las páginas y la publicación de artículos. Para el E-Commerce se implementó una aplicación en donde se pueden ingresar productos, crear categorías y subcategorías, ver las órdenes que se han creado y los pagos que se han realizado los clientes y otra aplicación donde se administra el registro de los clientes.

La tienda virtual se desarrolla para servir de base a cualquier tipo de comercio electrónico, por lo cual carece de contenido. Se escogió la empresa Enciso Systems para implementar el proyecto semilla, en esta implementación se generaron páginas y artículos con contenido de la empresa y se publicaron algunos productos de merchandising.

Palabras clave: Tienda virtual, Django, CMS, E-Commerce.

3. Planteamiento del problema

Antes de la emergencia sanitaria COVID-19, las ventas virtuales no tenían mucha aceptación por las personas, ya que preferían ir hasta el establecimiento y escoger los productos que querían adquirir, una de las principales razones fue la sensación de inseguridad al realizar un pago en línea y el temor a que no llegara el producto.

A inicio de la pandemia del COVID-19, los gobiernos decretaron cuarentena obligatoria, por lo cual las personas se vieron obligadas a realizar comprar en línea, provocando un aumento acelerado del comercio electrónico. Esto llevó a las empresas y negocios a actualizar o implementar tiendas virtuales, generando una oportunidad de negocio para las empresas de desarrollo de software.

Las tiendas virtuales son sitios web utilizados para el comercio electrónico, estos sitios web siguen la estructura de HTML5 (Header, body, footer). Por este motivo se puede decir que las tiendas virtuales son iguales y lo que las diferencia es su contenido y las tecnologías utilizadas para su desarrollo. Si se aprovecha esta semejanza el tiempo que se emplea para el diseño de un sitio web, se podría reducir el tiempo de desarrollo.

Durante todo el año la empresa Enciso Systems ha venido recibiendo propuestas de diferentes clientes para el desarrollo de un E-Commerce de acuerdo a sus necesidades. Teniendo en cuenta lo anterior, la empresa Enciso Systems quiere dar solución a la siguiente pregunta: ¿Cómo diseñar una herramienta que aproveche estas semejanzas de las tiendas virtuales y poder usarlas como base para desarrollar diferentes proyectos de forma rápida y eficaz?

4. Objetivos

4.1 Objetivo General

Desarrollar un proyecto semilla basado en tecnología Django de propósito general.

4.2 Objetivo Específicos

- Realizar el levantamiento de los requerimientos funcionales y no funcionales para establecer la arquitectura del proyecto.
- Integrar un CMS con el proyecto semilla realizado en Django.
- Desarrollar un proyecto semilla que contenga los componentes básicos de una tienda virtual.
- Implementar una pasarela de pago genérica.
- Desplegar el proyecto semilla en una tienda de prueba.

5. Justificación

El mejoramiento de procesos de software se centra en el perfeccionamiento de los procesos para mejorar la calidad de los productos, el rendimiento, la utilidad y la efectividad de los procesos. Esto permite a una organización de software mejorar continuamente sus capacidades para proporcionar servicios de calidad en forma competitiva [1].

A medida que las organizaciones crecen, existe una creciente necesidad de convertir el conocimiento personal en conocimiento organizacional. Por su parte, los conocimientos y experiencias que los miembros de los equipos de proyectos crean y adquieren durante el desarrollo de diferentes proyectos constituyen un valioso activo para las organizaciones que buscan mejorar sus prácticas y procesos de software [2].

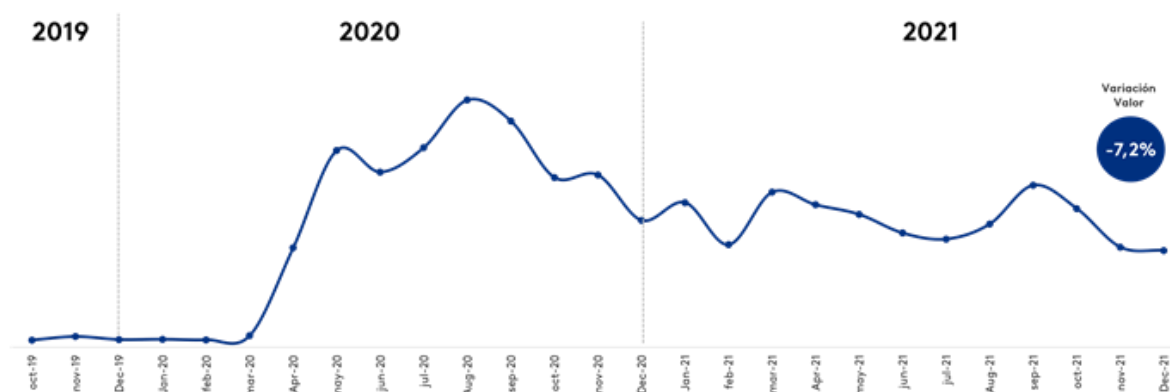
Por estas razones, la empresa Enciso Systems pretende realizar, en modalidad de pasantía, el desarrollo de un proyecto semilla que permita al equipo de trabajo contar con una herramienta que les ayuda a editar y administrar contenido, sin la necesidad de modificar código HTML o empezar un proyecto desde cero. Esto con el fin de mejorar los procesos y tiempos de producción de futuros proyectos relacionados al comercio electrónico.

6. Marco teórico

El mundo de las aplicaciones web es cada vez más conocido y accesible por la mayoría de las personas, son notables los enormes avances tecnológicos que la web día a día presenta y que llegan como herramientas para facilitar el trabajo de personas, empresas y hasta grandes organizaciones [3]. Por eso en estos últimos años las innovaciones de internet y las comunicaciones han aportado para que las aplicaciones web y los negocios mantengan una relación más directa y permita que el denominado comercio electrónico o E-commerce se desarrolle y llegue a un mayor número de consumidores.

La emergencia sanitaria (COVID-19) provocó un aumento acelerado del comercio electrónico, de hecho, estadísticas de Kantar (empresa británica de investigación de mercado), reflejaron que en el 2020 la penetración del E-Commerce durante la primera semana de confinamiento aumentó en un 100% y durante la cuarta semana alcanzó un 300% [4]. En la ilustración 1, podemos observar cómo fue el crecimiento de las ventas electrónicas antes, durante y después de la pandemia del covid-19.

Ilustración 1. Aumento del comercio electrónico del 2019 al 2021. Crédito: Kantar



Una Tienda Virtual es un lugar en internet en el que se puede vender las 24 horas del día diferentes clases de productos de manera parecida a como se los haría en una tienda tradicional. Debido a que las tiendas virtuales poseen un programa que les brinda la posibilidad de mostrar descripciones e imágenes de diferentes productos de manera amigable sin necesidad de la interrelación humana, además de ofrecer la alternativa de realizar pagos en línea [5].

También, es necesario mencionar que, al encontrarse en internet, una tienda virtual tiene la posibilidad de recibir clientes de todo el mundo, lo que genera la oportunidad de ofrecer sus productos a un mercado global y permite a los negocios presentar una imagen más profesional y confiable al momento de ser buscados a través de internet por los consumidores [6].

6.1. Django

Es un framework web de Python de alto nivel que fomenta el desarrollo rápido, el diseño limpio y pragmático. Creado por desarrolladores experimentados, se encarga de gran parte de la molestia del desarrollo web, por lo que puede concentrarse en escribir sus aplicaciones sin necesidad de reinventar la rueda. Es gratis y de código abierto [7].

Una de las partes más poderosas de Django es su interfaz de administración. Lee los metadatos del modelo para proporcionar una interfaz rápida y centrada en el modelo donde los usuarios de confianza puedan administrar el contenido del sitio [8].

6.2. Git

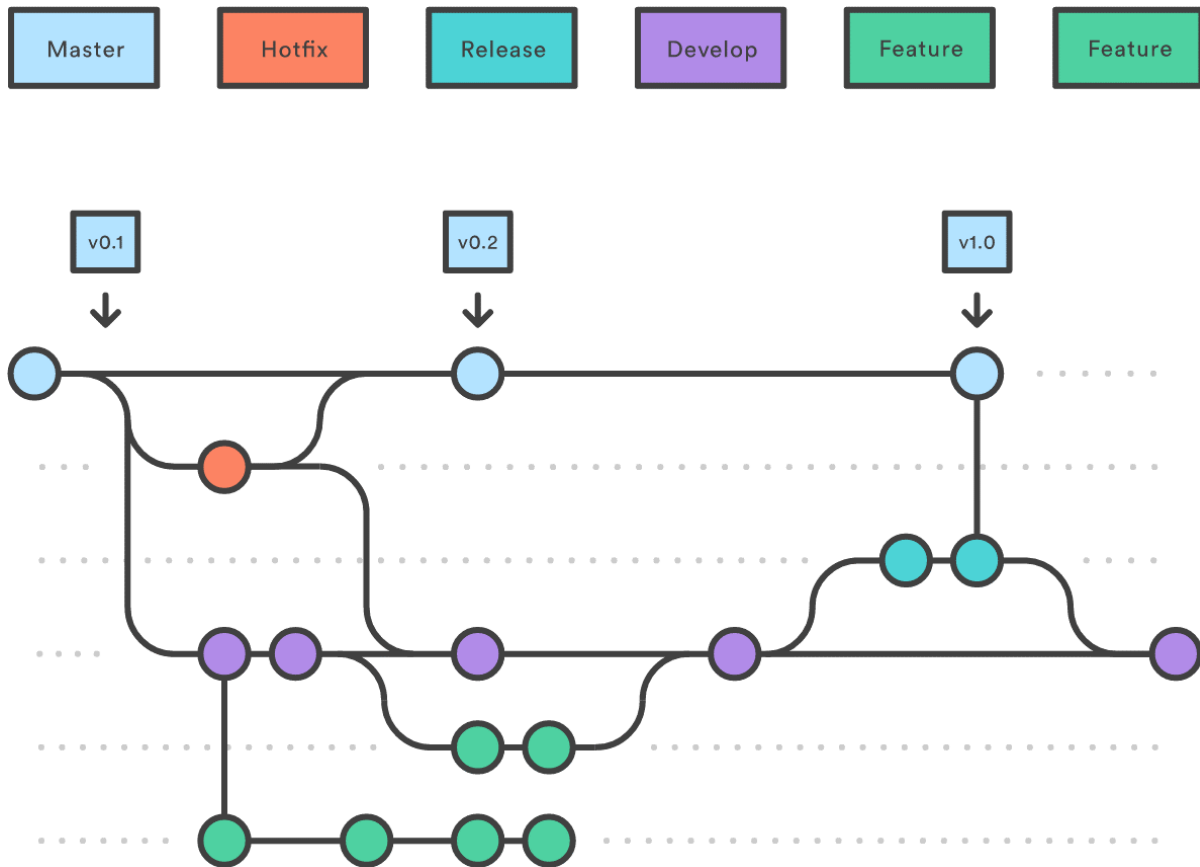
Git es un sistema de control de versiones distribuido que puede manejar desde proyectos pequeños hasta proyectos muy grandes, con facilidad. Git es fácil de aprender, ocupa poco espacio y funciona muy bien. Lo que lo convierte en una herramienta ideal para el seguimiento y control de diversos proyectos de desarrollo [9].

Git se encarga de registrar los cambios en uno o varios archivos del proyecto, lo que permite un control más organizado sobre los archivos del proyecto. También permite restaurar el estado anterior de un archivo después de realizar uno o varios cambios [10].

6.2.1 Flujo de trabajo en Git.

Git utiliza el flujo de trabajo llamado Gitflow, propuesto originalmente por Vicent Driessen [11]. Este se basa en la ramificación y permite realizar cambios en el código principal, pero no se reflejan hasta que se realiza una fusión. Estos cambios se realizan en una rama de desarrollo de la que se pueden crear ramas hija para los cambios menores. Las fusiones pueden realizarse mezclando las ramas hijas en la rama desarrollo y pasar los cambios de esta rama a la rama principal [10], como se muestra en la ilustración 2.

Ilustración 2. Flujo de trabajo en Git. Representación visual de Gitflow. Crédito: Atlassian



Este flujo de trabajo permite tener ramas separadas reservadas para partes específicas en el desarrollo.

- Master. La rama master siempre es el código de producción más reciente. El código experimental no pertenece aquí.
- Develop. La rama develop contiene todos los últimos desarrollos. Estos cambios de desarrollo pueden ser prácticamente cualquier cosa, pero las características más grandes se reservan para sus propias ramas.
- Hotfix. Las ramas de hotfix son para la corrección de errores menores, que no pueden esperar hasta la siguiente versión. Las ramas de hotfix salen de master y se fusionan de nuevo en master y en develop.
- Release. Las ramas de liberación se utilizan para liberar nuevos desarrollos de develop a master.
- Feature. Las ramas de features se reservan para las características más grandes. Estas se desarrollan específicamente en las ramas designadas y se integran con el desarrollo cuando están terminadas [11].

6.3. Docker

Docker es una tecnología de virtualización de contenedores. Es como una máquina virtual muy ligera. Empleada para empaquetar, transportar y ejecutar una aplicación como un código ligero [12].

Es decir, con Docker se puede empaquetar una aplicación y sus dependencias en un contenedor virtual que se puede ejecutar en cualquier servidor Linux. Esto ayuda a permitir la flexibilidad y portabilidad en el medio donde la aplicación se puede ejecutar, ya sea en instalaciones físicas, la nube pública o privada, entre otras [13].

6.4. Bootstrap

Bootstrap es un framework desarrollado por los creadores de la conocida red social Twitter y lanzado en 2011. Este marco fue creado utilizando los estándares de desarrollo web más recientes, como HTML5, CSS3 y jQuery.

El objetivo de Bootstrap es proporcionar una herramienta que permita a los desarrolladores de software crear páginas o aplicaciones web que sean visibles para los usuarios en cualquier dispositivo electrónico, como ordenadores de escritorio, tabletas, ordenadores portátiles y dispositivos móviles. Esta característica se conoce como diseño adaptable o responsivo, y pretende ofrecer una visualización adecuada en cada dispositivo utilizando un único diseño web [14].

7. Metodología y área de estudio

La metodología de desarrollo de software con la que se desea llevar a cabo el proyecto es la metodología Scrum, debido a que se realiza una gestión regular basada en resultados tangibles entregados de forma periódica, generando así calidad en la construcción del software a través de una retroalimentación continua a un ritmo de trabajo sostenible.

SCRUM es un marco de trabajo iterativo e incremental para el desarrollo de proyectos y se estructura en ciclos de trabajo llamados Sprint. Estos son iteraciones de 1 a 4 semanas, y se suceden una detrás de otra. Al comienzo de cada Sprint, el equipo multifuncional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. Al final del Sprint, el equipo lo revisa con los interesados en el proyecto, y les enseña lo que han construido. El equipo obtiene comentarios y observaciones que se pueden incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint, es decir que estén realizados [15].

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen

todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad [16].

7.1. Eventos de Scrum

- **PRODUCT BACKLOG:** a. Es la pila del producto; surge de la reunión entre el Development Team (DT), Product Owner (PO) y Scrum Master (SM) para definir los requerimientos del producto.
- **Sprint:** El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado” utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior.
- **SPRINT PLANNING:** a. De los requerimientos obtenidos, se eligen los que harán parte de la pila de tareas para el sprint (Sprint Backlog), detallando cada una de estas para que el DT tenga todo claro y se acepte bajo la definición de “Ready”.
- **SPRINT EXECUTION:** Esta da lugar, como lo indica su nombre, a la ejecución de sprint, es decir, al trabajo como tal que se va a realizar durante el lapso establecido.
 - **Daily Meeting:** Cada día dentro del sprint, se realiza una reunión diaria para contemplar los siguientes interrogantes:
 - ¿Qué se hizo el día anterior?
 - ¿Qué inconvenientes tuvieron?
 - ¿Qué se hará hoy?
- **SPRINT REVIEW:** a. Bajo la definición de “Done”, al final del sprint se tiene un hito considerado como el incremento funcional del producto en desarrollo. i. Acta de entrega del incremento con sus especificaciones (según se solicite). b. Se realiza una revisión para verificar el avance en el desarrollo, identificando qué faltó por cumplir, qué se procede a hacer y, en algunas ocasiones, determinar si el incremento pasa o no a producción. i. Documento de registro con los ítems mencionados anteriormente.
- **SPRINT RETROSPECTIVE:** Al final de cada sprint se realiza una actividad de retrospectiva para evaluar al equipo de desarrollo. Es liderada por el Scrum Master con el fin de identificar qué se ha hecho bien, qué no se debe seguir haciendo y qué se podría mejorar [17].

8. Ejecución

El primer paso para el desarrollo de la pasantía, fue analizar el funcionamiento de diferentes E-Commerce, esto permitió establecer las vistas, información y funcionalidades que tendría el E-Commerce. También se realizó un análisis de CMS existentes para entender el funcionamiento para el desarrollo del sistema de gestión de contenido.

De acuerdo a lo analizado se investigaron las tecnologías necesarias para el desarrollo del frontend y el backend del proyecto. En función de los requisitos iniciales, se eligieron las herramientas y tecnologías más adecuadas. También se dispuso de tiempo para la adaptación y el aprendizaje antes de empezar oficialmente el desarrollo de las funcionalidades y configuraciones del proyecto.

En la fase de adaptación fue necesario el aprendizaje de las siguientes tecnologías:

- Django: Para el desarrollo de los modelos, views y template del E-commerce y el CMS.
- Bootstrap: Para el diseño de los template del sitio web.
- Git: Para realizar un control de las versiones del proyecto y mantener un respaldo del código fuente en un repositorio en GitHub administrado por la empresa.

8.1. Desarrollo de las prácticas

Antes de iniciar la pasantía, se realizó una reunión liderada por el ingeniero Javier Enciso y dos miembros del equipo de Enciso Systems, en la cual se socializaron temas de interés sobre la pasantía. En esta reunión inicialmente el ingeniero realizó una breve presentación de la empresa, su funcionamiento interno, los objetivos, el equipo de trabajo y los proyectos en los que se encuentra trabajando.

También se explicó el objetivo del proyecto y lo que se quiere lograr, las tecnologías con las que se quiere trabajar, también hubo sugerencias por parte de los miembros de la empresa y la asignación de algunas tareas iniciales. Una de las indicaciones realizadas en esta reunión fue el empleo de Bootstrap 4 para realizar el diseño de los templates, así como el uso de Docker para realizar el despliegue en el servidor de Digital Ocean.

Al finalizar la reunión realice una presentación personal exponiendo mis habilidades y las expectativas acerca del desarrollo de la pasantía. También me enviaron un correo de invitación al repositorio que se encuentra alojado en GitHub.

8.1.1 Descripción de la duración de la pasantía

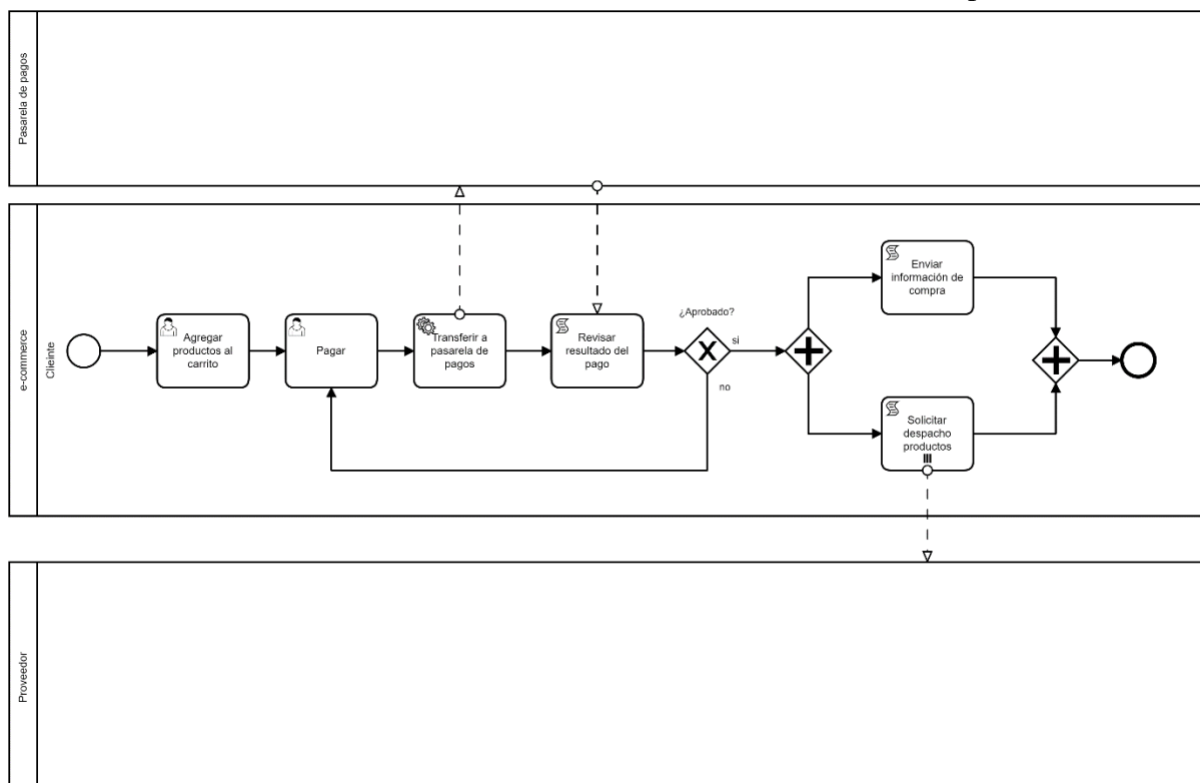
Primer mes

Al inicio de la pasantía, la empresa dio un tiempo de adaptación y se asignaron pequeñas tareas para entender el funcionamiento de Django, con esto se comprendió de una mejor manera la arquitectura modelo, vista, template que se maneja en el framework. También se aprovechó el tiempo para practicar y aprender las tecnologías que se necesitan para el desarrollo del proyecto, una de esas prácticas fue realizar ejercicios prácticos del uso de git para entender el flujo de trabajo que se maneja en el proyecto.

Aparte de estas tareas y ejercicios prácticos, se realizó una capacitación de BPMN, donde se revisó documentación de BPMN y se siguieron algunos tutoriales, esto con el objetivo de realizar el modelo de procesos del E-Commerce.

Al terminar el primer mes, se estableció el modelo de procesos del E-Commerce, el cual se realizó teniendo en cuenta los roles que participan, las actividades y su secuencia para realizar una compra, como se muestra en la ilustración 3.

Ilustración 3. Modelo BPMN del e-commerce. Crédito: Propio.



Segundo mes

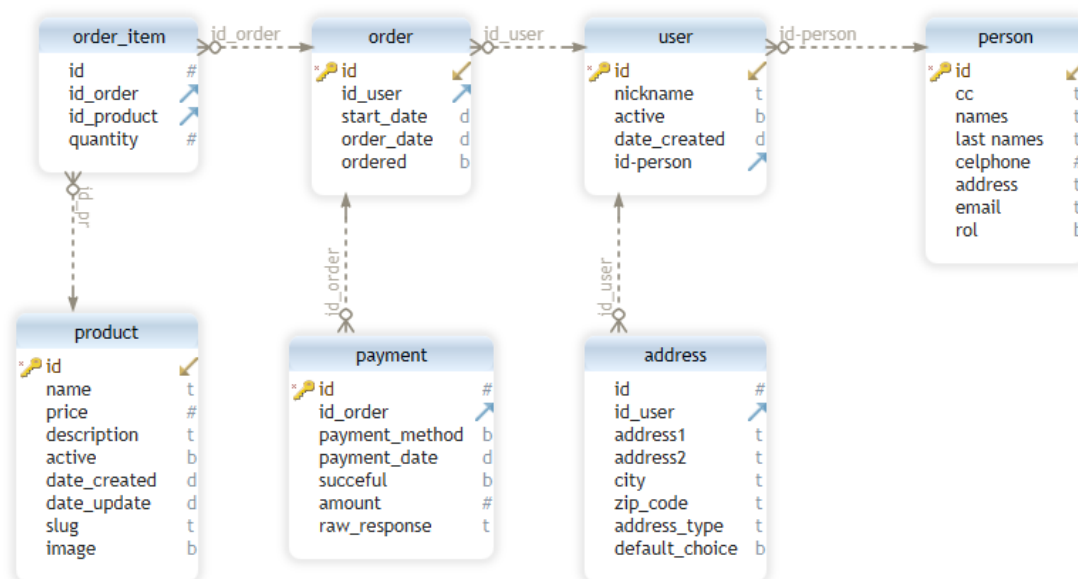
Durante la primera semana del segundo mes se realizaron mockups y así poder una idea de cómo iban a quedar las vistas del e-commerce y la información que se iba a mostrar en cada una de ellas, estos mockups fueron creados teniendo en cuentas el diagrama de procesos, obteniendo como resultado el diseño de las siguientes vistas.

- Home
- Listado de productos.
- Detalle del producto.
- Carrito de Compras.
- Login.

Gracias al estudio para realizar el modelo de procesos y el desarrollo de los mockups se hizo el levantamiento de los requisitos funcionales con mayor facilidad. Obteniendo como resultado el documento de requerimientos que será anexado a este documento, en el cual se especifica cada uno de estos.

En las siguientes semanas se trabajó en el desarrollo del modelo de datos como se muestra en la ilustración 4. Este modelo de datos es la base para el desarrollo de los modelos en Django, ya que en proceso de desarrollo se realizaron algunas modificaciones.

Ilustración 4. Modelo de datos del e-commerce. Crédito: Propio.



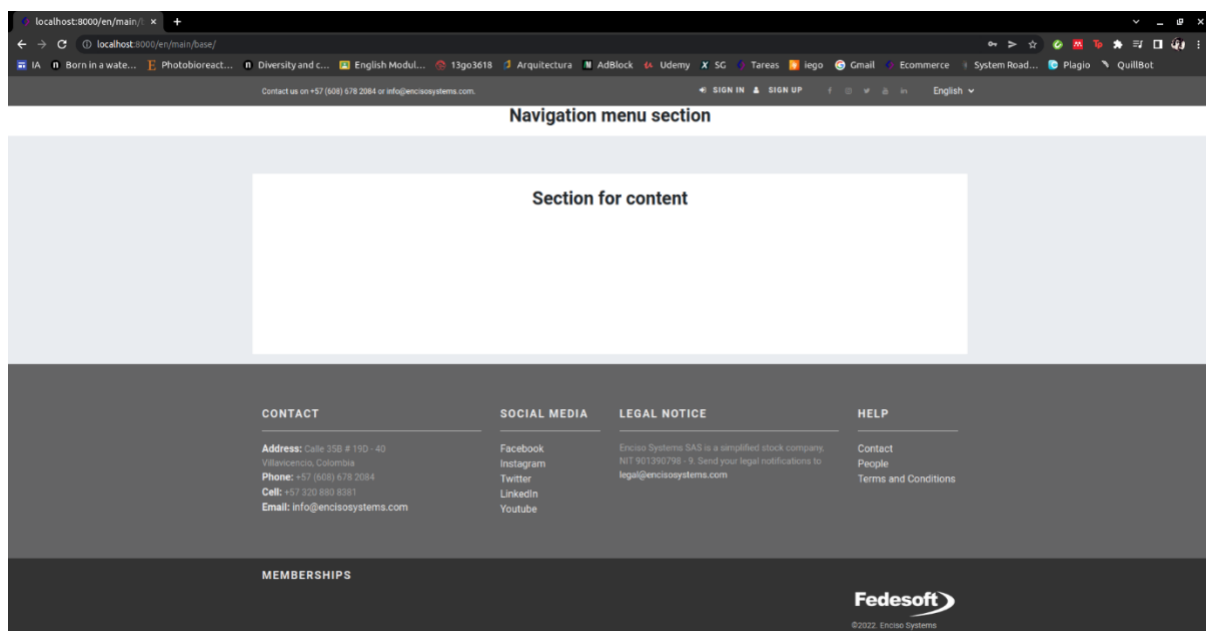
Al terminar el mes, se creó la aplicación shop en el proyecto, en esta aplicación se desarrollaron los modelos teniendo en cuenta el modelo de datos propuesto, estos modelos se registraron en el admin de Django para poder visualizar y administrar los datos que se registran en cada uno de estos modelos.

Tercer mes

En el tercer mes de la pasantía se empezó a trabajar en el frontend, para el desarrollo de los template se utilizó el sistema de herencia de plantillas de Django, este sistema permite crear un template base el cual contiene todas las partes comunes del sitio web y definir bloques que los template hijos van a sobrescribir, como se muestra en la ilustración 5.

Ilustración 5. Template que se utilizará como base para las vistas del e-commerce.

Crédito: Propio.



Se hizo uso del framework Bootstrap 4 para darle los estilos CSS al template base y posteriormente en los otros templates. Después de tener el template base se procedió a desarrollar el template Home y la vista Home, esta vista se encarga de renderizar la información que se muestra en la página de inicio. Una vez terminada la página de inicio se desarrollaron los templates y vistas para la página listar producto, detalle del producto, login, sign up y el perfil de usuario.

En las reuniones semanales de la empresa se iban mostrando los avances y a la vez se recibían nuevas indicaciones y correcciones del trabajo realizado, al terminar una funcionalidad, los cambios pasaban por una revisión y una vez eran aprobados se enviaban a la rama master para posteriormente subirlos a producción. En la tercera semana del tercer mes ya se había subido a producciones las funcionalidades que permitían por parte del administrador registrar productos, administrar los usuarios registrados y administrar las órdenes, por parte del usuario podía registrarse en el e-commerce, iniciar sesión, ver la lista de productos, ver los detalles de cada producto y agregarlos al carrito.

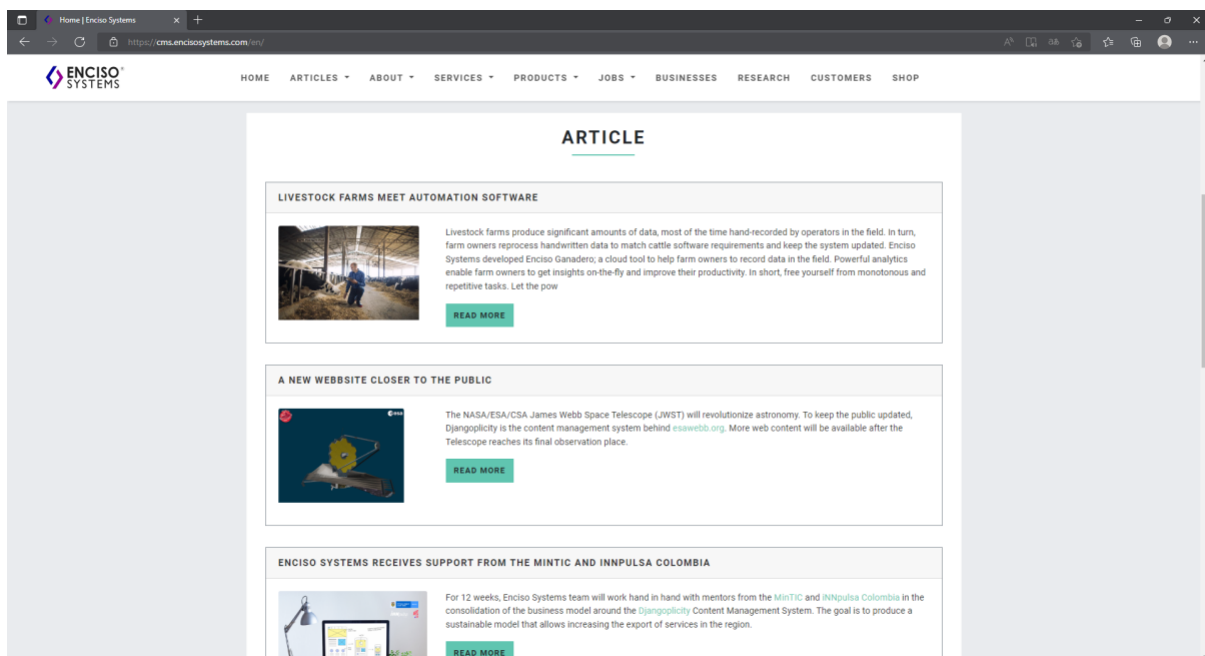
En la cuarta semana del tercer mes, se implementó la aplicación de menús, con sus respectivos modelos, vistas y templates. Esta aplicación es una de las funcionalidades que solicitó el ingeniero Javier Enciso para el CMS, la cual permite generar y administrar menús de navegación desde el admin de Django.

Cuarto mes

En este mes se trabajó en las otras dos funcionalidades solicitadas por el Ingeniero Javier Enciso para el CMS, estas funcionalidades permitirían crear, administrar y publicar páginas y artículos. Para ello fue necesario crear dos aplicaciones una llamada pages y la otra articles, en cada una de ellas se desarrollaron los modelos, template y vistas que utilizarían cada una. Después de que las aplicaciones fueran revisadas, los cambios se enviaron a la rama master y posteriormente se subieron a producción.

Una vez los cambios se aplicaron en producción, se asignó la tarea de crear páginas con información de la empresa, algunas de estas páginas son el About Us, Corporate Identity Contact, entre otras. Después de terminar de crear la paginas solicitadas, se continuó con la migración de unos artículos que tiene la empresa Enciso Systems, como se muestra en la ilustración 6.

Ilustración 6. Listado de artículos de Enciso Systems Crédito: Propio.



En la tercera semana de este mes se realizó la integración de la pasarela de pago, la pasarela que se escogió implementar fue la de PayPal, para esto se revisó la documentación de PayPal para entender su funcionamiento, los parámetros que se le tienen que enviar y como recibir los datos de confirmación del pago. Para la integración fue necesario utilizar el script que proporciona PayPal para poder agregar el botón de pago, y desarrollar el view que se encargaría

de guardar los datos del pago en el modelo de pagos. Después de realizar las pruebas de integración y ver que estaba funcionando correctamente se procedió a subir esta nueva funcionalidad a producción.

Por último, se procedió a verificar que toda la información que fue agregada desde el admin de Django, que se estuviera cargando correctamente en cada una de sus vistas, después de todo el proceso el sitio web quedó disponible para ser usada y administrada por el equipo de la empresa.

8.2 Implementación

Para tener una mejor organización y control de las tareas asignadas se hizo uso de un tablero de proyectos en GitHub, en donde se puede observar las tareas pendientes, las que están en progreso, las revisiones en proceso y las que han sido aprobadas y las tareas que han sido finalizadas.

Como se mencionó en la metodología se hizo uso del flujo de trabajo propuesto por git, esto permitió realizar las tareas o cambios mediante ramas que heredaban de Develop, estos cambios se subían al repositorio mediante commits y posteriormente se realizaba un Pull Requests en el que se realizaba una descripción de los cambios realizados, este Pull requests era revisado y aceptado por un miembro del equipo de la empresa. Después de terminar una tarea o requerimiento, este se subía a producción, para esto se fusiona la rama Develop con master para poder realizar el despliegue y posteriormente se verifica que no hubiera errores.

9. Resultados

9.1. Requerimientos

En la primera fase de la pasantía se realizaron reuniones con el ingeniero Javier Enciso y algunos miembros de la empresa, en estas reuniones se hablaron sobre los objetivos, las funcionalidades que se esperaban, las tecnologías a utilizar, entre otros. Después de tener claro los objetivos, se pasó a una fase de capacitación de BPMN donde se repasaron conceptos, componentes, funciones y buenas prácticas para realizar el modelo de procesos del e-commerce, este modelo se realizó teniendo en cuenta el comportamiento que tiene un cliente o usuario al momento de querer adquirir un producto. Este proceso ayudó a identificar las vistas básicas e información que debería tener el sitio web.

Después de presentar el modelo de procesos del e-commerce en las reuniones semanales de la empresa, el siguiente paso fue realizar los mockups del sitio web, teniendo en cuenta las vistas que se identificaron al momento de realizar el modelo de procesos, los mockups que se realizaron fueron.

- Vista de login.
- Vista de registro.
- Vista Home.
- Vista para listar los productos.
- Vista para ver los detalles del producto seleccionado.
- Vista del carrito de compras.
- Vista de la pasarela de pago.

Una vez realizado el diagrama de procesos y los mockups, se procedió a realizar el documento de requerimientos funcionales y no funcionales del sistema, donde se describen cada uno de ellos. Como se observa en las tablas 1 y 2.

Tabla 1. Requisitos funcionales del sitio web (E-Commerce). Crédito: Propio.

Requerimiento	Descripción
Registro de usuario	<ul style="list-style-type: none"> El E-Commerce deberá contar con un formulario de registro, que le permita al usuario ingresar nombre de usuario, contraseña y correo electrónico. Se debe desarrollar un formulario donde el usuario pueda registrar sus datos personales, como nombres y apellidos, número de teléfono, cédula, dirección de residencia y fecha de nacimiento.
Login	<ul style="list-style-type: none"> Desarrollar el registro y la validación del nombre de usuario y contraseña creados en el registro para poder iniciar sesión.
Sección de usuario	<ul style="list-style-type: none"> Se debe desarrollar la vista mi cuenta, donde el usuario podrá visualizar y actualizar sus datos personales. El usuario podrá visualizar las órdenes realizadas con su respectivo número de referencia, fecha, costo y el estado del pago.
Visualización de productos.	<ul style="list-style-type: none"> El E-Commerce deberá contar con una vista donde el usuario pueda ver los productos. El usuario podrá listar los productos por una categoría específica. Los productos se deben listar y visualizar con una imagen, nombre y precio del producto.
Detalle del producto	<ul style="list-style-type: none"> En la vista detallada del producto el usuario podrá ver la imagen o imágenes del producto, la descripción de las características, el nombre, el precio del producto. Se deberá mostrar el método o métodos de pago con los que cuenta el e-commerce. Se debe integrar un formulario donde el usuario pueda ingresar la cantidad de productos que quiera adquirir. En esta vista debe haber un botón que le permita al usuario añadir el producto al carrito de compras.
Carrito de compras	<ul style="list-style-type: none"> Dentro del carrito el usuario podrá ver de forma ordenada los ítems que ha ido agregando, podrá agregar más cantidades o eliminar el ítem. El carrito debe mostrar el precio de los ítems de acuerdo a las cantidades y una suma total de todo lo que ha sido agregado. En un único proceso el usuario podrá proceder a realizar el pago de todos los productos agregados.
Visualización de artículos.	<ul style="list-style-type: none"> El sitio web deberá contar con una vista que permita listar los artículos o noticias publicados por la tienda. Los artículos o noticia se podrán listar por categorías

Tabla 2. Requerimientos funcionales del CMS. Crédito: Propio.

Requerimiento	Descripción
Módulo de Menú	<ul style="list-style-type: none"> El administrador podrá crear menús root, los cuales serán relacionados con los menús ítem. El administrador podrá crear menús ítems, los cuales contarán con un nombre, un link, se podrán relacionar a otro menú ítem. El administrador podrá actualizar o eliminar menús y menús ítem que se hayan creado.
Módulo de paginas	<ul style="list-style-type: none"> El administrador podrá generar diferentes secciones, las cuales contarán con un nombre, el diseño del template y un título. El administrador podrá crear paginas informativas, donde podrá poner información de la empresa., las páginas deben contar con un título, el contenido y un link. El administrador podrá actualizar o eliminar las páginas que haya creado.
Módulo de artículos	<ul style="list-style-type: none"> El administrador podrá crear categorías y subcategorías que se relacionarán con los artículos o noticias que se creen. El administrador podrá crear artículos o noticias, los cuales contarán con nombre, categoría, imagen, título, resumen, contenido del artículo. El administrador podrá actualizar o eliminar los artículos que se hayan creado. Los artículos se deberán listar en la página de inicio. Se debe desarrollar una vista donde se pueda ver los detalles del artículo seleccionado.
Módulo de productos	<ul style="list-style-type: none"> El administrador podrá crear categorías y subcategorías que se relacionarán con los productos. El administrador podrá ingresar el nombre, la imagen, el precio, las unidades disponibles y la descripción del producto. El administrador podrá actualizar o eliminar los productos que se hayan ingresado. Se debe desarrollar una vista donde se listen los productos y las categorías. Se debe desarrollar una vista donde se pueda ver los detalles del producto seleccionado.
Módulo de usuarios.	<ul style="list-style-type: none"> El administrador puede ver los usuarios que se hayan registrado en el sitio web. El administrador podrá crear usuarios y dar privilegios de administración. El administrador podrá actualizar la información del usuario. El administrador podrá eliminar los usuarios registrados.

Los requerimientos no funcionales que se identificaron son los siguientes:

- Disponibilidad: El sistema debe permanecer operativo el 99% de las veces en que un usuario intente acceder.
- Facilidad de uso: El sistema debe proporcionar una interfaz de usuario fácil de usar y con información sobre las herramientas para mejorar y ser eficazmente receptivo.

- Seguridad: El sistema debe ser capaz de proporcionar seguridad contra cualquier inyección externa utilizando un sistema de seguridad por capas. La implementación de las funcionalidades de inicio de sesión del usuario también garantiza que el sistema sea seguro frente a personas no autorizadas.
- Rendimiento del sistema: El tiempo de respuesta es muy bueno para un trabajo determinado. El sistema soporta un entorno multiusuario.
- Fiabilidad del sistema: El sistema será altamente fiable y genera todas las actualizaciones
- información en el orden correcto. La validación y verificación de los datos se realiza en cada etapa de la actividad. La recuperación del sistema también será rápida.

Al terminar de realizar el documento de requerimientos, se pasó a la creación y configuración de la arquitectura que tendría el proyecto, el proyecto tendría una configuración de Docker para contenerizar la aplicación desarrollada en Django, PostgreSQL para la creación de la base de datos y Django donde se desarrollaría el proyecto, aprovechando que Django nos permite separar las funcionalidades del proyecto por medio de aplicaciones, se establecieron las siguientes aplicaciones.

Para el CMS

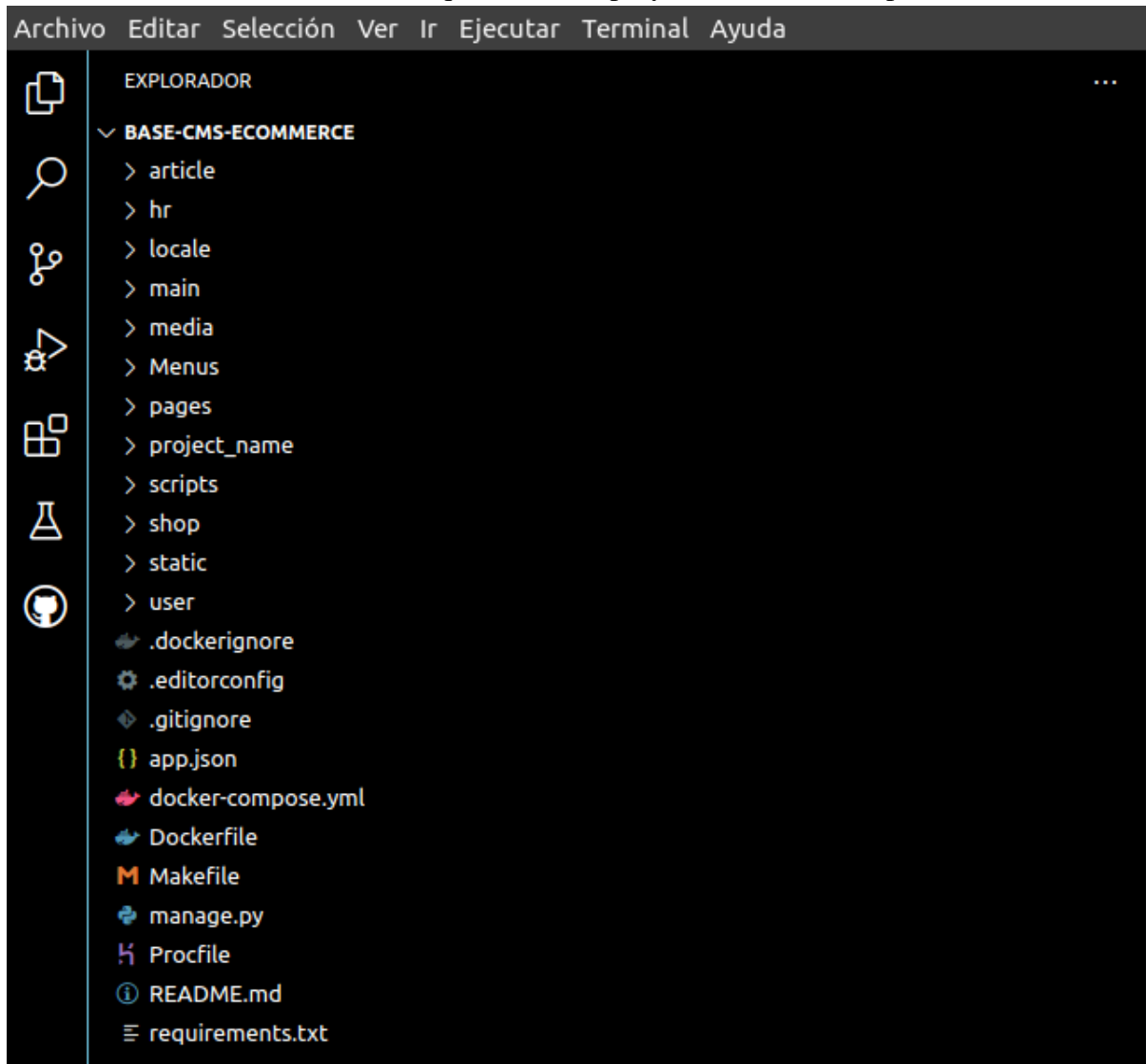
- Menu
- Pages
- Article

Para el E-Commerce

- Shop
- User

En la ilustración 7 se muestra cómo se observa la arquitectura del proyecto desde visual studio code.

Ilustración 7. Arquitectura del proyecto Créditos: Propio.



9.2. E-Commerce

Como se mencionó anteriormente, para el desarrollo de las funcionalidades del e-commerce se crearon dos aplicaciones, la primera que se creó fue la aplicación de shop y se establecieron los modelos que se necesitan para el almacenamiento de la información y posteriormente la creación de las tablas en la base de datos mediante las migraciones. Los modelos que se desarrollaron son los siguientes:

- Address para el registro de las direcciones de envío y facturación del cliente.
- Category para identificar a qué categoría pertenece el producto.
- Subcategoria para identificar a qué subcategoría pertenece el producto.
- Product. Para el registro de los productos.
- OrderItem. Para visualizar la relación entre un producto y una orden.
- Order. Registra la orden de compra que realiza un cliente.
- Payment. Para registrar los datos de pago de un cliente.

En la ilustración 8 se muestra la forma en la que se crea un modelo en Django. Después de terminar el desarrollo de los modelos, se registraron en el archivo admin.py y así poder visualizarlos en la interfaz de administración de Django, donde el administrador puede iniciar a registrar los productos de la tienda.

Ilustración 8. Campos, atributos y funciones del modelo OrderItem. Crédito: Propio.

```
class OrderItem(models.Model):
    """
    This model is the relation between the producto and the order.
    """
    order = models.ForeignKey("Order", verbose_name = _('Order'), related_name='items', on_delete=models.CASCADE)
    product = models.ForeignKey(Product, verbose_name = _('Product'), on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(verbose_name = _('Quantity'), default = 1, help_text=_("Quantity you want to buy"))

    class Meta:
        verbose_name = _('OrderItem')
        verbose_name_plural = _('OrderItems')

    def __str__(self):
        """ Return quantity and title of product"""
        return f"{self.quantity} x {self.product.title}"

    def get_raw_total_item_price(self):
        return self.quantity * self.product.price

    def get_total_item_price(self):
        price = self.get_raw_total_item_price()
        return price
```

Al terminar de crear los modelos, se procedió a crear los templates listar productos, detalle de producto, carrito de compras. El desarrollo de estos templates se realizó con base en los mockups propuestos y las sugerencias del equipo de trabajo. En paralelo se realizó la comunicación entre los template y los modelos mediante las vistas y así poder mostrar la información en el sitio web. En las ilustraciones 9, 10 y 11 se muestra el resultado de los templates.

Ilustración 9. Resultado final del template listar productos. Crédito: Propio

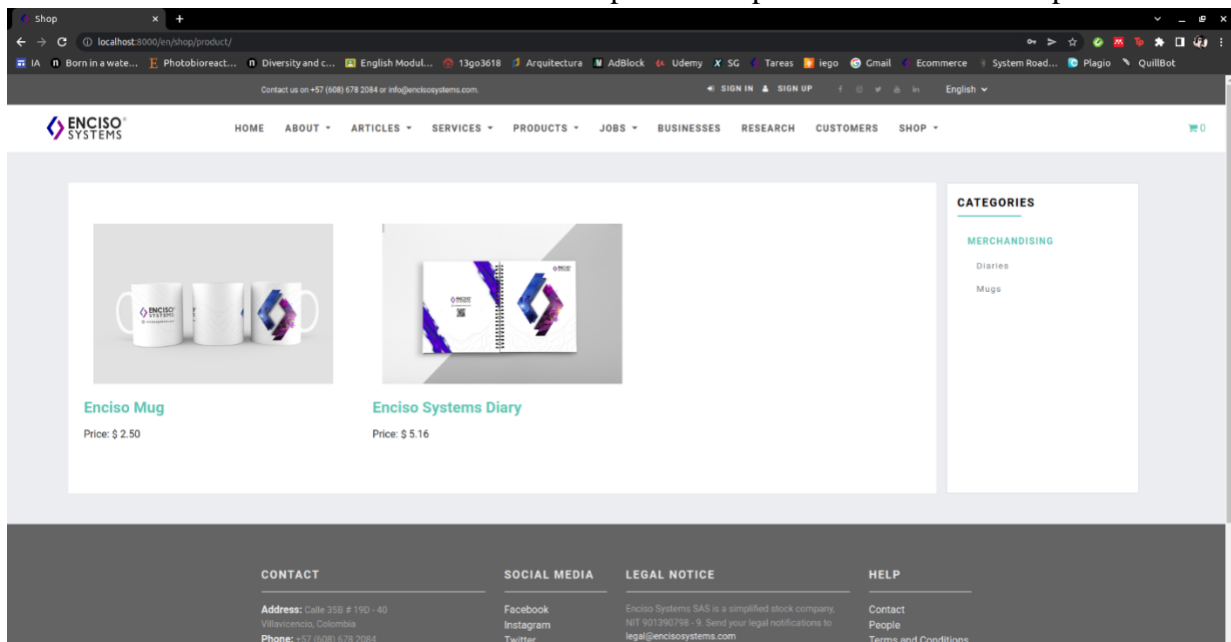


Ilustración 10. Resultado Final del template detalle del producto. Crédito: Propio

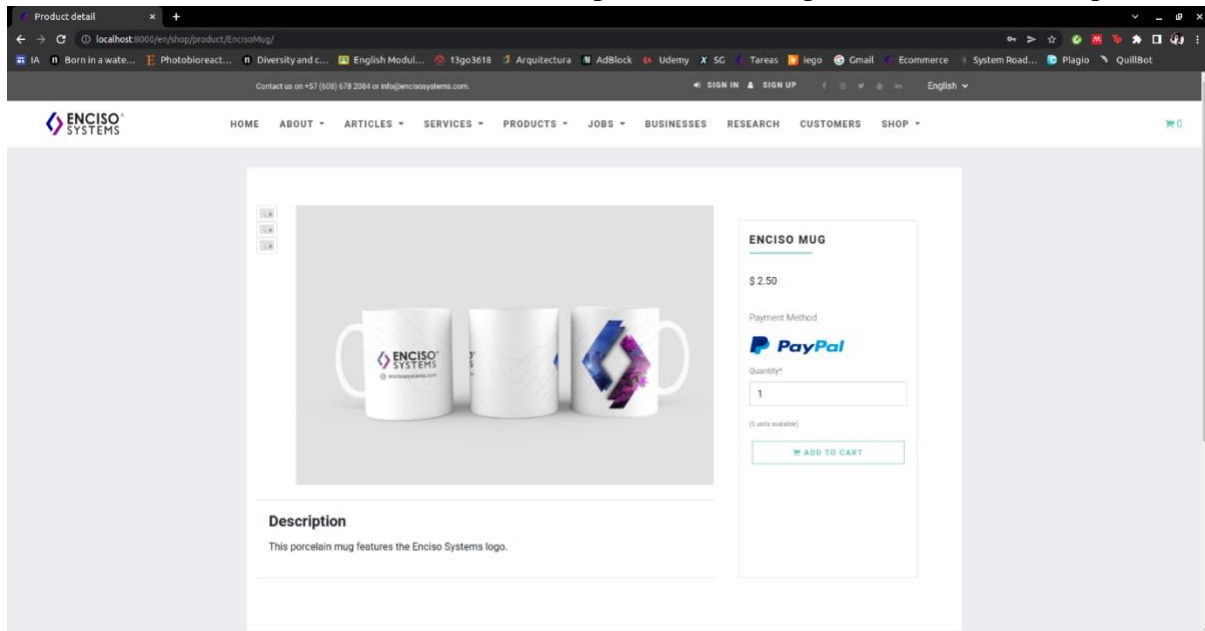
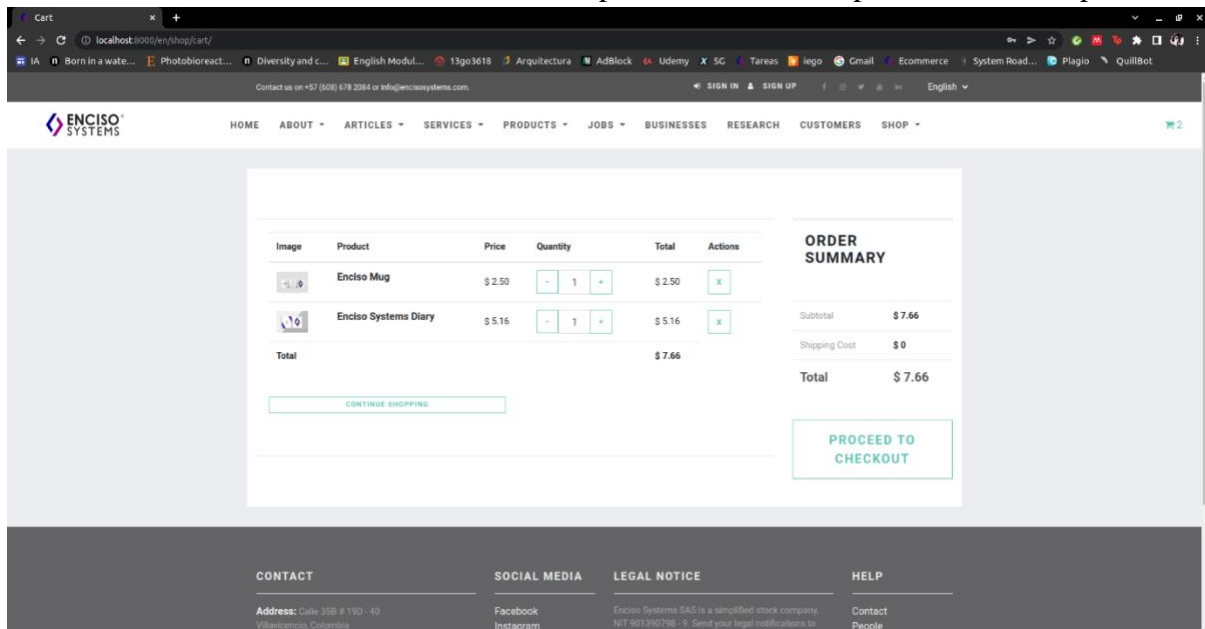


Ilustración 11. Resultado final del template carrito de compras. Crédito: Propio



Como se puede observar los templates heredan la estructura del template base que se mencionó anteriormente, lo único que se agrega es la información que se quiere mostrar, esto hace que el mantenimiento sea mucho más fácil y rápido.

Para la aplicación user se creó el modelo Person en el que se almacenarán los datos de registro de un usuario, como nombres, número de identificación, teléfono, entre otros. Después de crear el modelo se procedió a desarrollar los templates que permitirían iniciar sesión, realizar el registro de un usuario y ver el perfil del usuario donde podrá actualizar sus datos personales. En las ilustraciones 12, 13 y 14 se puede observar el resultado final.

Ilustración 12. Vista final del template Login. Crédito: Propio.

The screenshot shows a web browser window with the URL `localhost:3000/en/user/login/`. The page features the ENCISO SYSTEMS logo in the top left corner and a navigation menu with links: HOME, ABOUT, ARTICLES, SERVICES, PRODUCTS, JOBS, BUSINESSES, RESEARCH, CUSTOMERS, and SHOP. A shopping cart icon with '0' items is in the top right. The main content area is a light gray box with the title 'LOGIN' in bold. Below the title are two input fields for 'Username' and 'Password'. A green button labeled 'SIGN IN!' is positioned below the password field. Below the button, the text 'Not registered yet?' is displayed. Further down, a green link says 'Register now! It is easy and done in 1 minute and gives you access to special discounts and much more!'. The footer is a dark gray bar with four sections: CONTACT (Address: Calle 35B # 150 - 40), SOCIAL MEDIA (Facebook), LEGAL NOTICE (Enciso Systems SAS is a simplified stock company...), and HELP (Contact).

Ilustración 13. Vista final donde un usuario puede registrarse en el e-commerce. Crédito: Propio.

The screenshot shows a web browser window with the URL `localhost:3000/en/user/signup/`. The page layout is identical to the login page, featuring the ENCISO SYSTEMS logo, navigation menu, and shopping cart. The main content area is a light gray box with two columns. The left column is titled 'NEW ACCOUNT' and contains four input fields: 'Username*', 'Password*', 'Password confirmation*', and 'Email*'. A green button labeled 'REGISTER!' is at the bottom of this column. The right column is titled 'LOGIN' and contains two input fields: 'Username' and 'Password'. A green button labeled 'SIGN IN!' is at the bottom of this column. The footer is the same dark gray bar as the login page.

Ilustración 14. Vista del perfil de usuario. Crédito: Propio.

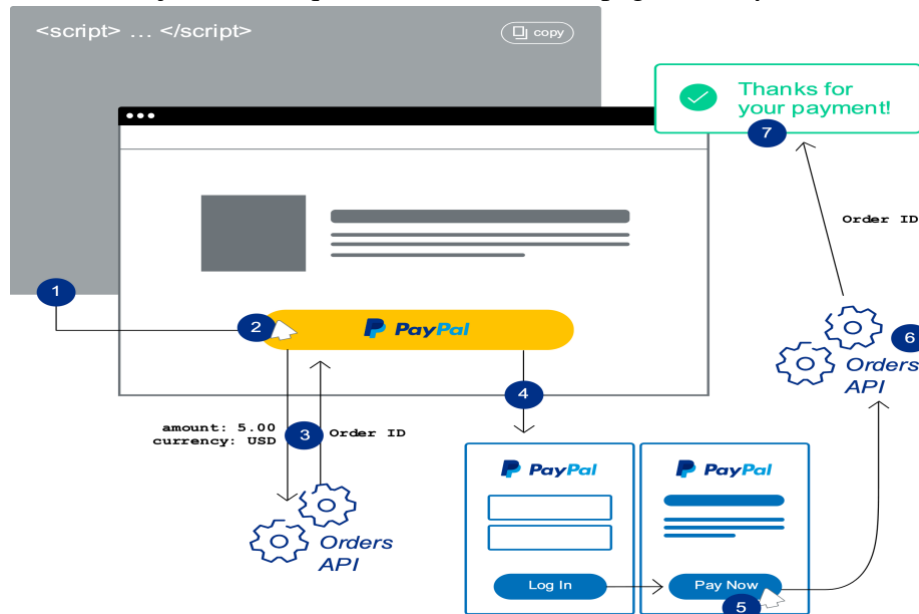
The screenshot shows a web browser window with the URL 'localhost:3000/myUser/dialmy/'. The page is titled '@DIALMY | DIEGO MARTINEZ'. It features a navigation bar with links: HOME, ABOUT, ARTICLES, SERVICES, PRODUCTS, JOBS, BUSINESSES, RESEARCH, CUSTOMERS, and SHOP. The main content area is divided into two sections: 'PERSONAL DETAILS' and 'CUSTOMER SECTION'. The 'PERSONAL DETAILS' section contains form fields for Name (Diego), Last Name (Martinez), Email (dialmy_94@hotmail.com), Age (18), Identification Card (1121912983), Phone number (3013786252), and Address (calle 24# 20a 08 este, Pinarilla). A 'SAVE CHANGES' button is located below the address field. The 'CUSTOMER SECTION' contains links for Settings, My account, and Logout. Below this, there is a 'YOUR ORDERS' section with a table header: Reference No., Date, Amount, and Payment Status. The table currently shows 'You haven't made any purchases'.

Al terminar el desarrollo de las aplicaciones shop y user se procedió a realizar la integración de PayPal como pasarela de pago.

9.3. Pasarela de Pagos

Para la integración de PayPal como pasarela de pago, fue necesario crear una cuenta en PayPal y entrar al dashboard de desarrollador para poder obtener la llave pública y privada con la que nos comunicaremos con la pasarela de pago, posteriormente se creó el template de payment utilizando el script que proporciona PayPal para la implementación del botón de pagos, este botón llama la API de órdenes de PayPal para configurar una transacción e iniciar la experiencia de pago, donde el usuario aprobará el pago, en la ilustración 15 podemos ver el flujo que realiza la pasarela de pago para confirmar un pago.

Ilustración 15. Flujo Técnico que realiza el botón de pagos de PayPal. Crédito: PayPal



La pasarela de pagos permite pagar con una cuenta de PayPal o con tarjeta de crédito, como se muestra en la Ilustración 16. Al realizar el pago se redirigirá al usuario a una página de confirmación y agradecimiento, como se muestra en la Ilustración 17.

Ilustración 16. Botones de pago con PayPal. Crédito: Propio.

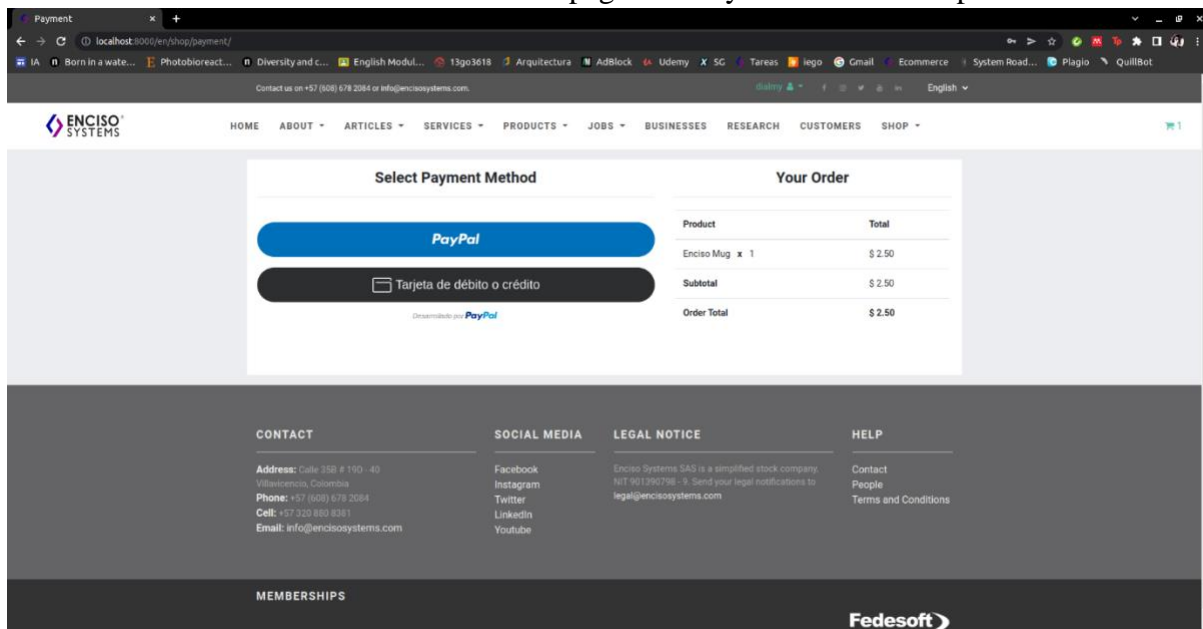
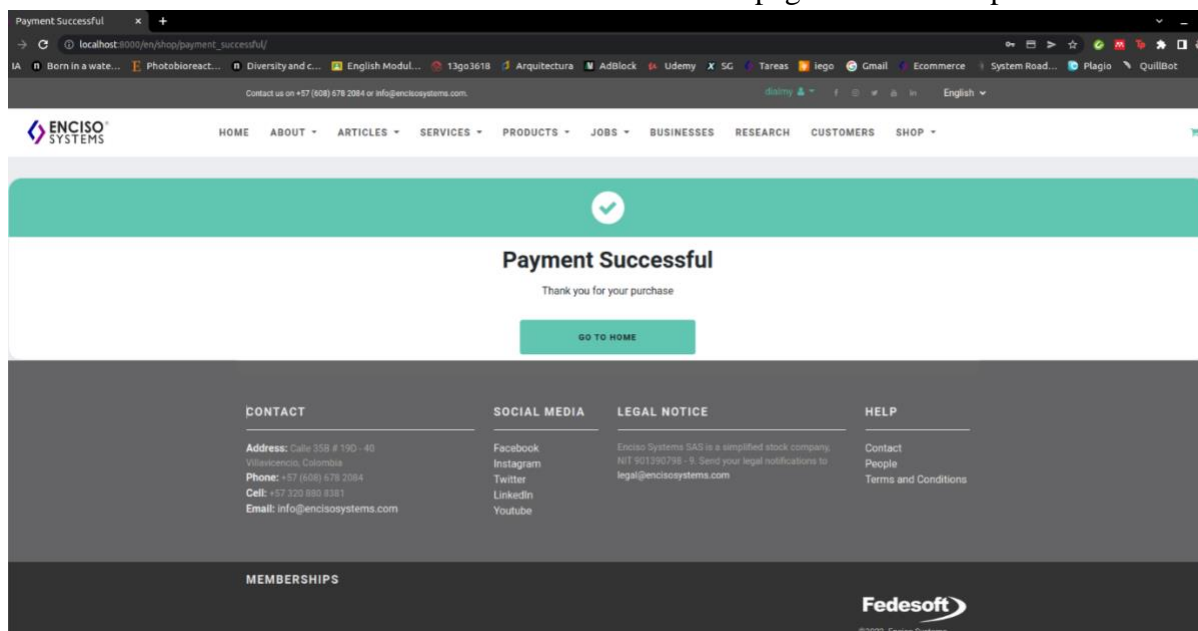
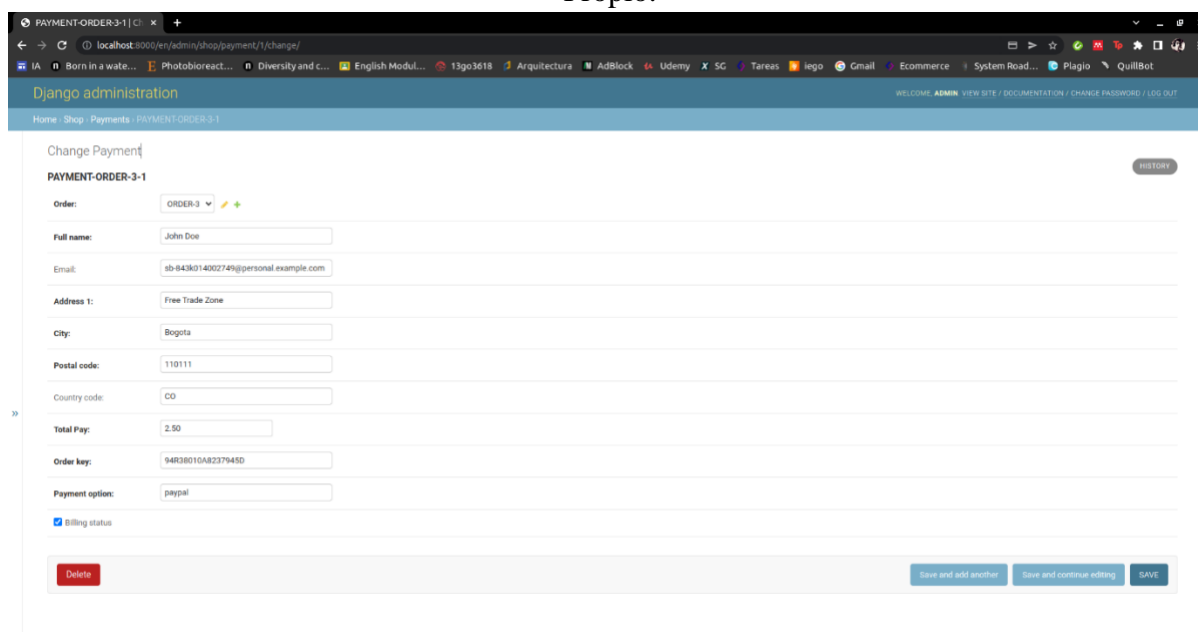


Ilustración 17. Vista de la confirmación del pago. Crédito: Propio.



También se desarrolló una función que se encargará de capturar los datos de la orden que se crea en PayPal y estos datos son guardados en el modelo Payment para que el administrador pueda visualizarlos en el sitio de administración como se muestra en la ilustración 18.

Ilustración 18. Visualización de los datos del pago en el administrador de Django. Crédito: Propio.



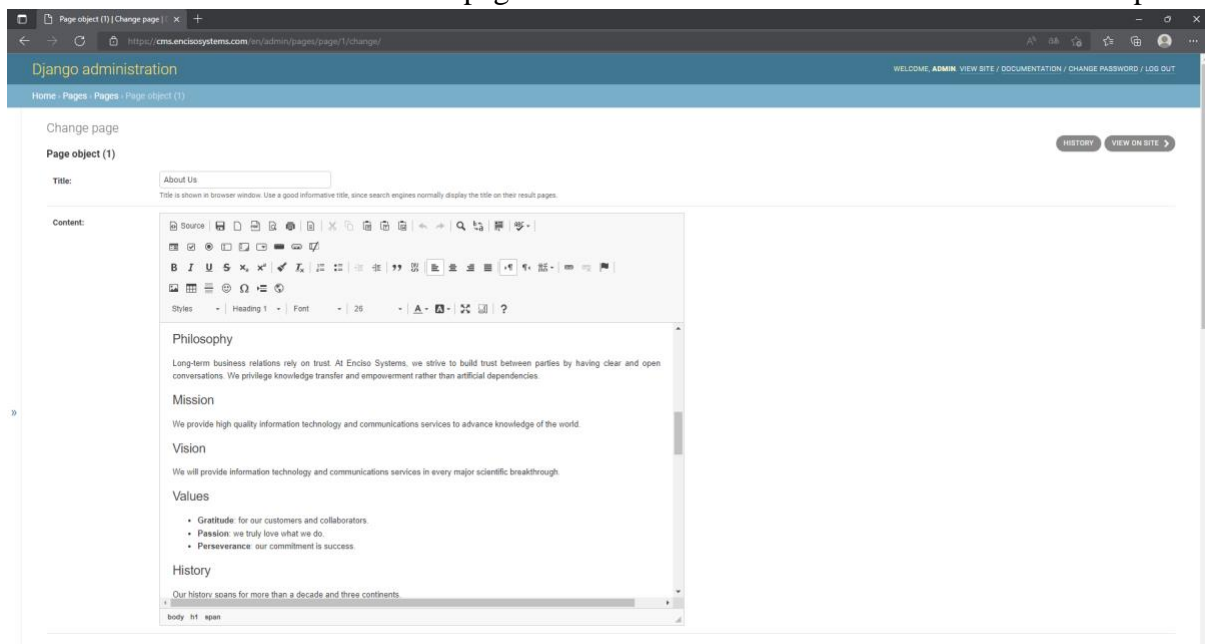
9.4. CMS

Las funcionalidades para el cms se desarrollaron de la misma manera que el e-commerce, se creó una aplicación para cada funcionalidad, esto le permite al administrador gestionar el menú de navegación, páginas y artículos desde el sitio de administración sin tener que tocar código html. La primera aplicación que se creó fue la pages y se crearon los siguientes modelos:

- Sections. Se utiliza para definir el template que se utilizará para generar la página.
- Links. Para generar un link de acceso a la página.
- Pages. Para crear y actualizar las páginas que se quieran crear.

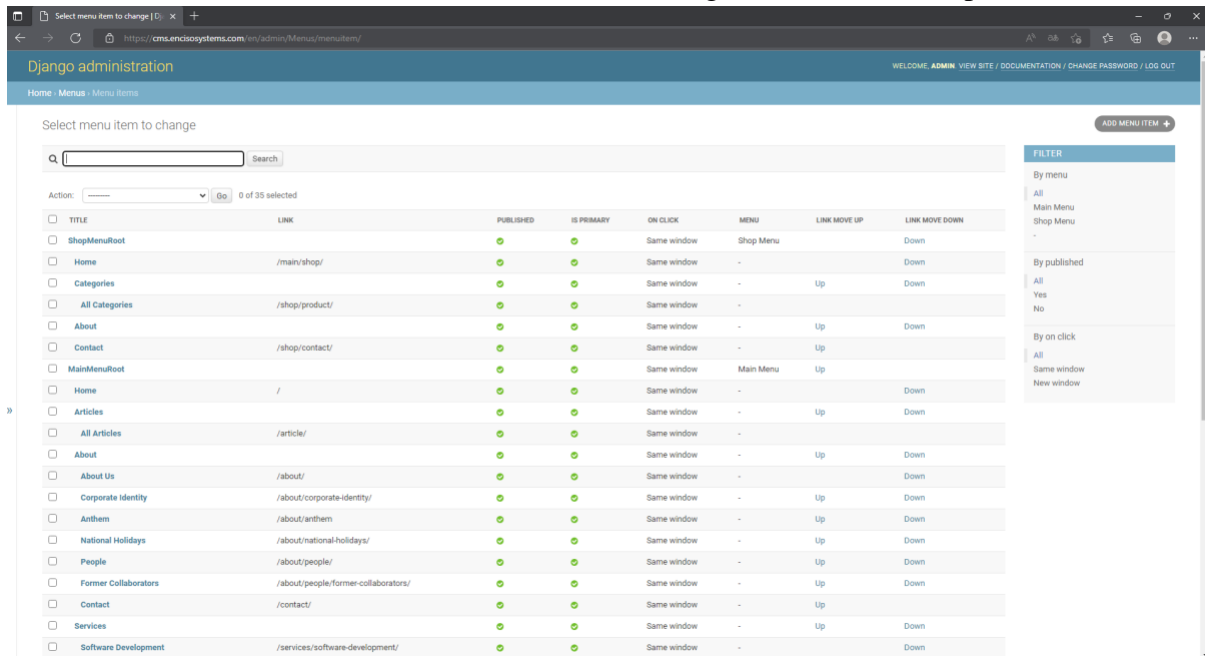
En la ilustración 19 se muestra cómo se crea una página desde el sitio de administración de Django.

Ilustración 19. Creación de una página desde el sitio de administración. Crédito: Propio



La siguiente aplicación que se creó fue la de menús, se crearon dos modelos, el modelo menú se encarga de agrupar una lista de menús y el modelo de menuitems se utiliza para generar la lista de menús como se muestra en la ilustración 20.

Ilustración 20. Lista del menú de navegación. Crédito: Propio.

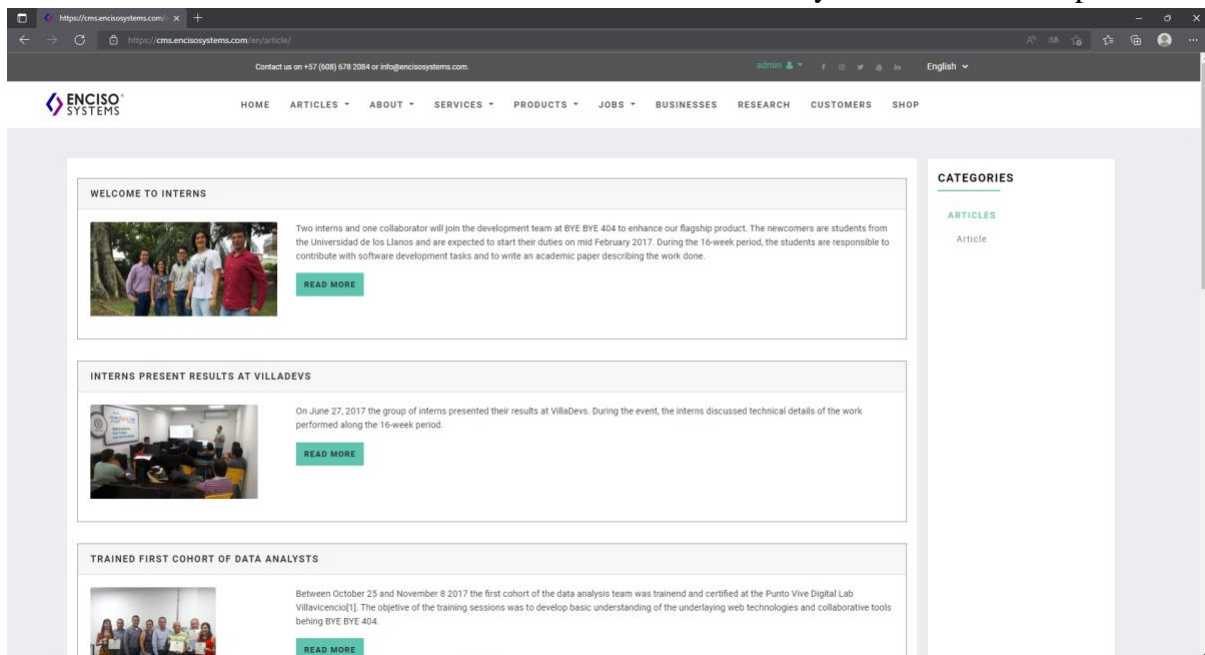


The screenshot shows the Django administration interface for menu items. The page title is "Select menu item to change". There is a search bar and a table of menu items. The table has columns: TITLE, LINK, PUBLISHED, IS PRIMARY, ON CLICK, MENU, LINK MOVE UP, and LINK MOVE DOWN. The table lists various menu items such as ShopMenuRoot, Home, Categories, All Categories, About, Contact, MainMenuRoot, Home, Articles, All Articles, About, About Us, Corporate Identity, Anthem, National Holidays, People, Former Collaborators, Contact, Services, and Software Development. A filter sidebar on the right allows filtering by menu, published status, and on-click behavior.

TITLE	LINK	PUBLISHED	IS PRIMARY	ON CLICK	MENU	LINK MOVE UP	LINK MOVE DOWN
<input type="checkbox"/> ShopMenuRoot		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	Shop Menu		Down
<input type="checkbox"/> Home	/main/shop/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		Down
<input type="checkbox"/> Categories		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> All Categories	/shop/product/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		
<input type="checkbox"/> About		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> Contact	/shop/contact/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	
<input type="checkbox"/> MainMenuRoot		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	Main Menu	Up	
<input type="checkbox"/> Home	/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		Down
<input type="checkbox"/> Articles		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> All Articles	/article/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		
<input type="checkbox"/> About		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> About Us	/about/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		Down
<input type="checkbox"/> Corporate Identity	/about/corporate-identity/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> Anthem	/about/anthem	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> National Holidays	/about/national-holidays/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> People	/about/people/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> Former Collaborators	/about/people/former-collaborators/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> Contact	/contact/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	
<input type="checkbox"/> Services		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-	Up	Down
<input type="checkbox"/> Software Development	/services/software-development/	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Same window	-		Down

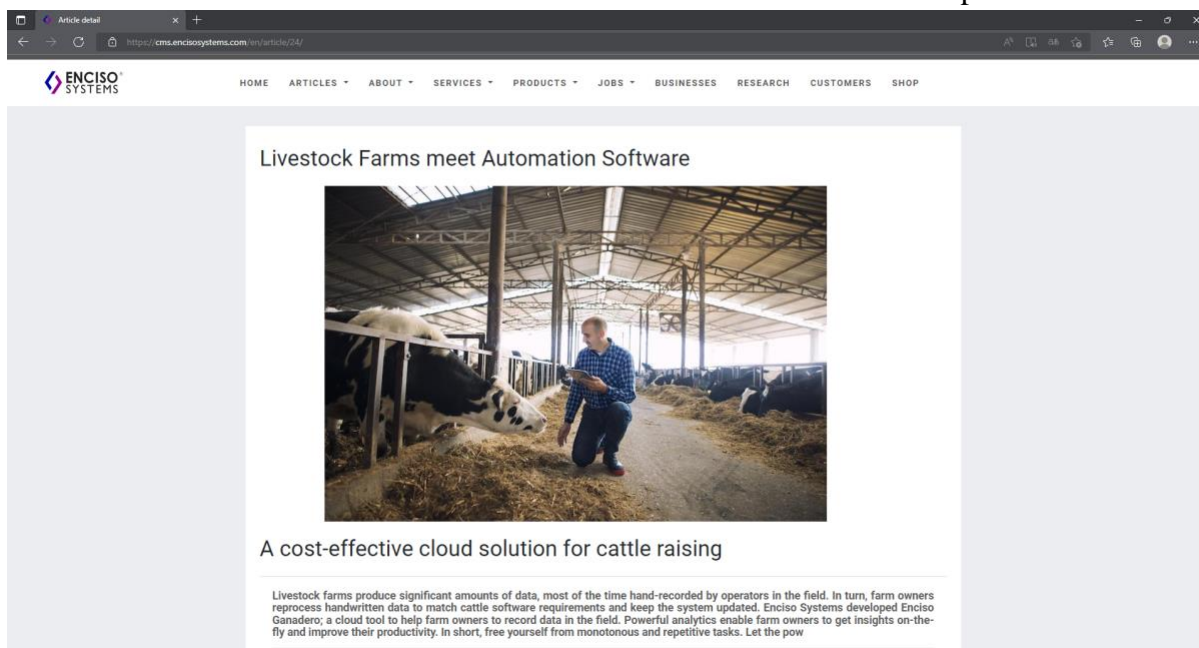
Por último, se creó la aplicación article, la cual tiene como funcionalidad la creación de artículos, en esta aplicación también se creó un template para listar los artículos y otro template donde se puede visualizar el artículo a detalle, también se desarrolló un modelo de categorías para agrupar los artículos en diferentes categorías. En la ilustración 21 se muestra el resultado final de esta funcionalidad.

Ilustración 21. Vista de listado de artículos de Enciso Systems. Crédito: Propio.



The screenshot shows the Enciso Systems website. The header includes the company logo and navigation links: HOME, ARTICLES, ABOUT, SERVICES, PRODUCTS, JOBS, BUSINESSES, RESEARCH, CUSTOMERS, and SHOP. The main content area features three article cards, each with a title, a thumbnail image, a brief description, and a "READ MORE" button. The articles are: "WELCOME TO INTERNS", "INTERNS PRESENT RESULTS AT VILLADEVS", and "TRAINED FIRST COHORT OF DATA ANALYSTS". A sidebar on the right titled "CATEGORIES" shows a list of articles under the "ARTICLES" category.

Ilustración 22. Vista detallada de un artículo. Crédito: Propio.



9.5. Despliegue

Para poner marcha el proyecto, se realizó la prueba con la empresa Enciso Systems en la que se pondría en funcionamiento el cms con la migración de páginas y artículos que tiene alojados en su página principal que está desarrollada en Drupal, con el objetivo de empezar a utilizar el producto obtenido en esta pasantía, para el e-commerce la empresa empezará a vender productos de merchandising, como mugs, libretas, gorras, entre otros.

El proyecto se encuentra contenerizado en Docker y cuenta con una configuración que permite realizar un despliegue sencillo en el servidor de producción. El proyecto se desplegó en un servidor de DigitalOcean donde Enciso Systems adquiere servicios de alojamiento en la nube. Para acceder al sitio web se puede realizar en el siguiente enlace: cms.encisosystems.com

10. Discusión

Hoy en día las personas han cambiado su forma de comprar productos, ahora prefieren adquirirlos de manera virtual, lo que a lleva a los negocios buscar nuevas formas de ofrecer sus productos porque no basta con tener un espacio físico, una de las soluciones es contar con una tienda virtual de manera que sus productos o servicios estén disponibles a sus clientes las 24 horas y así poder llegar a más clientes.

En la actualidad existen diferentes plataformas que le permiten a una persona o empresa crear su tienda virtual de manera sencilla, pero estos sitios no permiten integrar nuevas funcionalidades de manera fácil, lo que lleva a las personas a contratar empresas de desarrollo para obtener una tienda virtual de acuerdo a sus necesidades, pero este proceso es más demorado.

Enciso Systems ha identificado estos problemas y decidió desarrollar una herramienta que le permita a su equipo desarrollar tiendas virtuales de forma más rápida y así mismo sus clientes puedan obtener un producto de acuerdo a sus necesidades. Para cumplir con este objetivo fue necesario el uso de herramientas que permitieran un desarrollo ágil y escalable a futuro. Uno de los primeros pasos fue identificar el proceso que realiza un usuario para comprar un producto e identificar los requerimientos.

Para el desarrollo de la aplicación se escogió el framework Django, ya que permite un desarrollo rápido y flexible, cuenta con aplicaciones muy útiles, como el registro de usuarios, manejo de grupos y permisos, entre otros. Además, Django cuenta con una interfaz para acceso a la base de datos, que permite hacer consultas de manera muy sencilla. Se trabajó con PostgreSQL para el almacenamiento de los datos y con Docker para la contenerización del proyecto.

Esto permitió desarrollar una aplicación lista para la creación y administración de contenido, catálogo de productos, registro de usuarios y pagos en línea. Permitiendo le ha Enciso Systems ofrecer a sus clientes una solución más rápida y eficaz para la creación de sus tiendas virtuales.

11. Conclusiones

1. La realización de un análisis de diferentes sistemas, plantear y entender el proceso del sistema a desarrollar y una buena definición de requerimientos, fueron de suma importancia para establecer y asignar las funcionalidades a desarrollar, gracias a esta planificación las funcionalidades establecidas se realizaron de forma organizada, permitiendo entregar avances en las reuniones semanales de la empresa.
2. Se implementaron tres aplicaciones (menus, pages y articles) que permiten crear y administrar menús de navegación, páginas de contenido y la publicación de artículos de manera sencilla, permitiéndole a un usuario la administración del contenido de su sitio web sin necesidad de tener conocimientos en HTML o programación.
3. Como resultado final de la pasantía, se desarrolló una aplicación que le permite a un administrador crear, actualizar y eliminar productos, agruparlos por categorías y subcategorías, visualizar las órdenes generadas por los usuarios y los pagos realizados. Los usuarios pueden registrarse en el sitio web, ver el catálogo de los productos, ver la descripción y detalles de un producto en específico, agregar productos al carrito y realizar pagos virtuales.
4. La integración de la pasarela de pagos de PayPal, le permite al usuario terminar su proceso de compra realizando el pago ya sea desde su cuenta de PayPal o por medio de una tarjeta de crédito y le permite a la empresa recibir pagos de diferentes países.
5. El despliegue de la aplicación se realizó por medio de la tecnología Docker, permitiendo una fácil implementación del contenedor configurado en el droplet de DigitalOcean. A medida que se terminaba de desarrollar una funcionalidad, se realizaba el respectivo despliegue, lo que permitió al equipo de trabajo e involucrados realizar sugerencias.

12. Recomendaciones

- A lo largo del desarrollo del proyecto se tuvo que generar un campo a diferentes modelos para agregar contenido multimedia, debido a esto, si una imagen o video se quería utilizar en diferentes modelos se tenía que cargar varias veces generando un mayor uso de almacenamiento, por esta razón se recomienda implementar el modelo *Media* en el CMS. Para la gestión de contenido multimedia y de esta manera poder relacionar un archivo en diferentes modelos.
- Actualmente el proyecto tiene integrada la pasarela de pagos de PayPal que permite realizar el pago a través de una cuenta de PayPal o una tarjeta de crédito o débito, pero esta pasarela no realiza pagos con pesos colombianos, por lo que sería conveniente integrar una pasarela que permita integrar una pasarela de pago que admita la moneda local.

13. Bibliografía

- [1] A. García Rodríguez, Y. Trujillo Castañola, and A. Perdomo, “Optimización de estados en la mejora de procesos de software,” *Enl@ce Rev. Venez. Inf. Tecnol. y Conoc.*, vol. 13, no. 2, pp. 9–27, May 2016.
- [2] G. Lopez, L. M. Figueroa, P. Rosa, M. Marilena, and F. Carola, “Optimización de la calidad del Proceso Software con Gestión del Conocimiento,” 2012.
- [3] C. J. Motoche Quezada, “DESARROLLO DE UNA APLICACIÓN WEB DE COMERCIO ELECTRÓNICO B2C PARA LA PRESENTACIÓN Y VENTA DE PRODUCTOS DE SEGURIDAD,” Machala, 2016.
- [4] T. Pereda, “Estado del eCommerce 2021 y su crecimiento en situación de pandemia,” 28-Mar-2021. [Online]. Available: <https://lemon.digital/estado-del-ecommerce-2021/>. [Accessed: 11-May-2022].
- [5] C. D. Ramírez Jaramillo and R. M. Moreira Zambrano, “Desarrollo e implementación de una tienda virtual utilizando oscommerce, caso práctico tienda virtual para material eléctrico y telefónico de media y baja tensión para la empresa ORGATEC,” Universidad de las Fuerzas Armadas, Ecuador, 2014.
- [6] J. M. Franco Cedeño, “Desarrollo De Una Aplicación Web Para Comercio Electrónico B2C Utilizando Un Framework Java,” *Utmach*, 2017.
- [7] Django, “Conoce a Django.” [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 27-Jun-2021].
- [8] Django, “The Django admin site | Django documentation | Django,” *Documentation Django*. [Online]. Available: <https://docs.djangoproject.com/en/4.0/ref/contrib/admin/>. [Accessed: 06-May-2022].
- [9] Git, “Git.” [Online]. Available: <https://git-scm.com/>. [Accessed: 09-May-2022].
- [10] S. Chacon and B. Straub, *Pro Git*. Apress, 2009.
- [11] “LEARNING GIT,” *Rip Tutorial*. [Online]. Available: <https://riptutorial.com/Download/git.pdf>. [Accessed: 11-May-2022].
- [12] C. Anderson, “Docker,” *IEEE Softw.*, vol. 32, no. 3, pp. 102–105, May 2015.
- [13] Á. Iradier and I. Martínez, *Docker para DevOps de noob a experto*, vol. 1. tecnología espublico, 2021.
- [14] E. Barrera, “Desarrollo de un sistema web multiplataforma de gestión de selección del personal para el área de Talento Humano, utilizando el framework Bootstrap,” *Univ. del Azuay*, pp. 1–204, 2018.
- [15] S. I. Mariño and P. L. Alfonzo, “Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación,” *Sci. Tech. ISSN 0122-1701, Vol. 19, N° 4, 2014*, págs. 413-418, vol. 19, no. 4, pp. 413–418, 2014.
- [16] S. Ken and S. Jeff, “La Guía de Scrum,” 2016.
- [17] G. V. Rojas Galindo, “IMPLEMENTACIÓN DE LA METODOLOGÍA ÁGIL SCRUM PARA EL DESARROLLO DE UNA PLATAFORMA WEB CON TIENDA VIRTUAL,” Villavicencio, 2019.

14. Anexos

- Especificación de requerimientos de software. [Link](#)